The papermas package

H.-Martin Münch
<Martin.Muench at Uni-Bonn.de>

2011/08/22 v1.0h

Abstract

This LaTeX package allows to compute the number of sheets of paper needed to print a document as well as the mass of that printed version of the document, useful e. g. when sending it by mail to determine the postage. (The number of pages of a document can be determined with the pageslts package.)

Disclaimer for web links: The author is not responsible for any contents referred to in this work unless he has full knowledge of illegal contents. If any damage occurs by the use of information presented there, only the author of the respective pages might be liable, not the one who has referred to these pages.

Save per page about 200 ml water, 2 g CO_2 and 2 g wood:
Therefore please print only if this is really necessary.

Contents

1 Introduction

This package is kind of an add-on to the pageslts package, but because that already uses some resources and computing the number of sheets of paper or the paper mass probably is not needed so often, this was made into a separate package.

It allows to compute the number of sheets of paper needed to print a document (useful when the paper is running out) as well as the mass of that printed version of the document, useful e. g. when sending it by mail to determine the postage.

Warning/Disclaimer: The mass of (printer's) ink has to be added and that of envelope, address sticker, stamps,... Thus this is only an estimation without guarantee – do not sue me, if you have got to pay excess postage!

The name papermas is short for paper mass but written with only one s, because some software has problems with names with more than eight letters.
It is mass and gives a result in grammes [g], because the weight $F = m \cdot g$ (really $\overrightarrow{F} = m \cdot \overrightarrow{g}$) [N] would require the knowledge of the gravitational acceleration g (depending on place and time, in central Europe approximately $9.81\,\mathrm{m/s^2}$) and give a result in NEWTON, which probably is not very useful.

2 Usage

Just load the package placing

```
\usepackage[<options>]{papermas}
```

in the preamble of your LaTeX 2_ε source file (preferably after calling the pageslts package).
Because the pageslts package is used to get the total number of pages, please place a `\pagenumbering{...}` with appropriate argument (e. g. arabic, roman, Roman, fnsymbol, alph, or Alph) right behind `\begin{document}` (see documentation of pageslts package).
Now you can say

```
This document consists of $\arabic{pagesLTS.pagenr}$~pages.
When printing $\papermaspagespersheet$~pages on one sheet of
paper, $\papermassheets$~sheets will be needed. For
ISO~A~\papermasformat\ paper of $\papermasmasss \unit{g}\unit{m}^{-2}$
specific mass, the printout will have a mass of about
$\papermasstotal \unit{g}$.
```

to get e. g.

> This document consists of 101 pages. When printing 4 pages on one sheet of paper, 26 sheets will be needed. For ISO A 4 paper of $80\,\mathrm{g\,m^{-2}}$ specific mass, the printout will have a mass of about 130 g.

This information is also presented at the screen while compiling your document (look for papermas), in the log file (search for *** Paper mass ***), and can be found in the aux file – probably one does not want to have the information in the printed document.
One could use the (x)color package and

```
{\color{white} This document ... of about $\papermasstotal \unit{g}$.}
```

which does not show in the printed document (white background of the page assumed), but can be made visible on the screen be marking that text.

2.1 Options

options The papermas package takes the following options:

2.1.1 format

format The option `format` wants to know the ISO A... format of the paper used for printing, i. e. `format=4` means ISO A4 paper format (which is also the default).

2.1.2 masss

masss The option `masss` wants to know the specific (therefore the third `s`) mass of the paper used for printing in g/m^2. The default is `masss=80`, i. e. $80\,\mathrm{g}/\mathrm{m}^2$.

2.1.3 pagespersheet

pagespersheet The option `pagespersheet` wants to know, how many pages are to be printed on one sheet of paper. `pagespersheet=2` could mean duplex printing or printing two pages on one side of paper while keeping the back side blank. This does not influence the real printing process! So, if this number differs from the one chosen for printing, the result will be wrong, of course.

2.1.4 decimalsep

decimalsep The option `decimalsep` wants to know, what should be used for the decimal separator. In English this is ".", while in German it is ",". Enclose this in brackets, e. g. `decimalsep={.}` or `decimalsep={,}`. The default is ".". This is used for the mass of the printed document, and this value is given at the screen during compilation as well as in the `log` and `aux` files. Therefore something like `decimalsep={,\,}` would cause trouble there.

3 Alternatives

For determining the number of pages (not sheets of paper) instead of the pageslts package the alternatives listed in the description of that package could be used, but then the according code in this package would need to be changed (and also e. g. the `ifcounter` command used here).
With the totpages package optionally the number of sheets of paper needed to print the document can be computed, too.
(See pageslts documentation.)

(You programmed or found another alternative, which is available at CTAN:?
OK, send an e-mail to me with the name, location at CTAN:, and a short notice, and I will probably include it in the list above.)

About how to get those packages, please see subsection 6.1.

4 Example

```
⟨*example⟩
\documentclass[british,a4paper]{article}[2007/10/19]% v1.4h
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\usepackage{hyperref}[2011/04/17]% v6.82g
\hypersetup{%
 extension=pdf,%
 plainpages=false,%
 pdfpagelabels=true,%
 hyperindex=false,%
 pdflang={en},%
 pdftitle={papermas package example},%
 pdfauthor={Hans-Martin Muench},%
 pdfsubject={Example for the papermas package},%
 pdfkeywords={LaTeX, papermas, Hans-Martin Muench},%
 pdfview=Fit,%
 pdfstartview=Fit,%
 pdfpagelayout=SinglePage,%
 bookmarksopen=false%
}
\usepackage[pagecontinue=true,alphMult=ab,AlphMulti=AB,fnsymbolmult=true,%
            romanMult=true,RomanMulti=true]{pageslts}[2011/08/08]% v1.2a
%% These are the default options. %%
\usepackage[format=4,masss=80,pagespersheet=2,decimalsep={.}]{papermas}
%% These are the default options. %%
\listfiles
\begin{document}
\pagenumbering{arabic}

\section*{Example for papermas}
\markboth{Example for papermas}{Example for papermas}

This example demonstrates the use of package\newline
\textsf{papermas}, v1.0h as of 2011/08/22 (HMM).\newline
The used options were \texttt{format=4} (ISO~A4),
\texttt{masss=80} ($\unit{g}\unit{m}^{-2}$), and\newline
\texttt{pagespersheet=2} (pages per sheet of paper,
i.\,e. either duplex printing or\newline
printing two pages on one side of a sheet of paper with blank back side).\newline
(These are the default options.)\newline
For more details please see the documentation!\newline

\bigskip

This document consists of
\lastpageref{LastPages}~(\arabic{pagesLTS.pagenr})~pages.
When printing $\papermaspagespersheet$~pages on one sheet of
paper, $\papermassheets$~sheets will be needed. For
ISO~A~\papermasformat\ paper of $\papermasmasss \unit{g}\unit{m}^{-2}$
specific mass, the printout will have a mass of about
$\papermasstotal \unit{g}$.

\bigskip

\noindent Save per page about $200\unit{ml}$ water,
$2\unit{g}$ CO$_{2}$ and $2\unit{g}$ wood:\newline
Therefore please print only if this is really necessary.\newline
```

```
I do NOT think, that it is necessary to print THIS file, really\newline
(at least not after this page)!

\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Page \thepage
\newpage Last page \thepage.

\end{document}
⟨/example⟩
```

5 The implementation

We start off by checking that we are loading into LaTeX 2_ε and announcing the name and version of this package.

```
115 ⟨*package⟩
116 \NeedsTeXFormat{LaTeX2e}[2009/09/24]
117 \ProvidesPackage{papermas}[2011/08/22 v1.0h
118             Computes paper mass of a printout (HMM)]
119
```

A short description of the papermas package:

```
120 %% Allows to compute the number of sheets of paper
121 %% needed to print a document as well as the
122 %% mass of that printed version of the document,
123 %% useful e. g. when sending it by mail to determine the postage.
124 %% Warning/Disclaimer: Mass of (printer's) ink has to be added
125 %% and that of envelope, address sticker, stamps,...!
126 %% So, this is only an estimation without guarantee -
127 %% do not sue me, if you have got to pay excess postage!
128
```

For the handling of the options we need the kvoptions package of Heiko Oberdiek (see subsection 6.1):

```
129 \RequirePackage{kvoptions}[2010/12/23]% v3.10
```

For the total number of pages we need the pageslts package of myself (see subsection 6.1):

```
130 \RequirePackage{pageslts}[2011/08/08]% v1.2a
131 \RequirePackage{intcalc}[2007/09/27]%  v1.1; for intcalcPow
```

A last information for the user:

```
132 %% papermas may work with earlier versions of LaTeX and those
133 %% packages, but this was not tested. Please consider updating
134 %% your LaTeX and packages to the most recent version
135 %% (if they are not already the most recent version).
136
```

See subsection 6.1 about how to get them.

The options are introduced:

```
137 \SetupKeyvalOptions{family = papermas,prefix = papermas@}
138 \DeclareStringOption[4]{format}[4]%        paper foormat, ISO A...,
139 %%                                        default: (ISO A) 4
140 \DeclareStringOption[80]{masss}[80]%       specific mass of the paper,
141 %%                                        default: 80 (g/(m^2))
142 \DeclareStringOption[2]{pagesperssheet}[2]% number of pages per sheet,
143 %%                                        for duplex printing this is 2.
144 \DeclareStringOption[.]{decimalsep}[.]%    decimal separator,
145 %%            e. g. "." or ",": decimalsep={,} - brackets are needed!!!
146 %%            decimalsep={,\,} does not work for screen, aux, log output!
147
148 \ProcessKeyvalOptions*
149
```

unit We define a \unit command:

```
150 \gdef\unit#1{\mathord{\thinspace\mathrm{#1}}}%
151
```

Even if diverse commands are not defined yet, we do not want a LaTeX `Error: ... undefined`.

```
\@ifundefined{papermasstotal}{\gdef\papermasstotal{\textbf{??}}}{}
\@ifundefined{papermasstotal}{\gdef\papermasstotal{\textbf{??}}}{}
\@ifundefined{papermasformat}{\gdef\papermasformat{\textbf{??}}}{}
\@ifundefined{papermasmasss}{\gdef\papermasmasss{\textbf{??}}}{}
\@ifundefined{papermaspagespersheet}{\gdef\papermaspagespersheet{\textbf{??}}}{}
\@ifundefined{papermassheets}{\gdef\papermassheets{\textbf{??}}}{}

```

`\papermas@totmass` This is the internal command, which computes the total paper mass of the printed document.

```
\newcommand\papermas@totmass{%
  \newcounter{papermasA}% paper mass for ISO A...
  \setcounter{papermasA}{\papermas@format}% e. g. 4
```

We check whether `papermasA` has a resonable value:

```
  \ifnum \value{papermasA}<0%
    \PackageError{papermas}{Option format has no valid value}%
     {The format option of the papermas package\MessageBreak%
      only takes whole, non-negative numbers (0, 1, 2, 3,...),\MessageBreak%
      because this should be the paper format\MessageBreak%
      ISO A 0, 1, 2, 3,...\MessageBreak%
      Found instead: \papermas@format \MessageBreak%
     }
  \else%
```

`papermasA` has a resonable value. We introduce a new counter `papermasmasss` and initialize it with the value given in option `masss`, i. e. `\papermas@masss`.

```
    \newcounter{papermasmasss}% always 0
    \setcounter{papermasmasss}{\papermas@masss}% default: 80
```

Counters are integers, but the amount of the mass of a single sheet of paper in most cases is not an integer, therefore we multiply with 100 to get two digits behind the decimal separator.
(Later we need to devide by 100 again, of course.)

```
    \multiply \value{papermasmasss} 100 % default: 8000
```

We check whether `papermasmasss` has a resonable value, i. e. > 0:

```
    \ifnum \value{papermasmasss}<1%
      \PackageError{papermas}{Option masss has no valid value}%
       {The masss option of the papermas package\MessageBreak%
        only takes positive numbers,\MessageBreak%
        because this should be the mass per square meter\MessageBreak%
        of a single sheet of your paper.\MessageBreak%
        Found instead: \papermas@masss \MessageBreak%
       }
    \else
```

`masss` has a resonable value, and therefore also `\papermas@masss` and `papermasmasss`.

We check whether option `pagespersheet` has a resonable value, i. e. ≥ 1:

```
      \newcounter{papermasPPS}% is 0
      \setcounter{papermasPPS}{\papermas@pagespersheet}% default 2
      \ifnum \value{papermasPPS} < 1%
        \PackageError{papermas}{%
```

```
          The number of pages per sheet must be positive.}{%
          You cannot print less than one TeX page per sheet of paper.\MessageBreak%
          The value found was \papermas@pagespersheet .\MessageBreak%
          }
      \else
```

`pagespersheet` has a resonable value, and therefore also `\papermas@pagespersheet` and `papermasTmpA`.

We introduce a new counter `papermas@sheets` for the number of sheets printed and initialize it with the number of pages as computed by package pageslts, i. e. `pagesLTS.pagenr`.

```
        \newcounter{papermas@sheets}
        \setcounter{papermas@sheets}{\arabic{pagesLTS.pagenr}}%
```

When more than one page is printed on one sheet of paper, the number of sheets needed for printing is decreased:

```
        \divide \value{papermas@sheets} by \value{papermasPPS}%
```

`\divide` cuts off all digits behind the decimal separator, but if there are digits > 0, this means that there is an additional, last sheet, which is only partially covered with print (e. g. only one side of it for duplex printing an odd number of pages). In that case, we have to add one sheet of paper to the number of sheets needed.

```
        \newcounter{papermas@tmpn}
        \setcounter{papermas@tmpn}{\arabic{papermas@sheets}}%
        \multiply \value{papermas@tmpn} \value{papermasPPS}%
        \ifnum \value{papermas@tmpn}=\value{pagesLTS.pagenr}
          \relax
        \else
          \addtocounter{papermas@sheets}{1}%
        \fi
```

Now we can multiply the specific mass of 100 sheets with the number of sheets needed for printing:

```
        \multiply \value{papermasmasss} \value{papermas@sheets}
  % default:                  8000       (no default for this)
```

The result is in $\mathrm{g\,m^{-2}}$.
A sheet with format ISO A0 has a size of $1\,\mathrm{m}^2$,
a sheet with format ISO A1 has a size of $1\,\mathrm{m}^2 \cdot 2^{-1}$,
a sheet with format ISO A2 has a size of $1\,\mathrm{m}^2 \cdot 2^{-2}$,
..., and
a sheet with format ISO An has a size of $1\,\mathrm{m}^2 \cdot 2^{-n}$.

Therefore we compute $2^{\value{papermasA}}$ and divide the specific paper mass by that value:

```
        \divide \value{papermasmasss} by \intcalcPow{2}{\value{papermasA}}
  % default:               16000      /   2^(\value{papermasA})
```

We need to get the division by 100 and the digits after the decimal separator right:

```
        % for the example 297 is used
        \newcounter{papermas@tmpm}
        \setcounter{papermas@tmpm}{\arabic{papermasmasss}}%   m:297 n:    o:  p:  q:
        \setcounter{papermas@tmpn}{\arabic{papermasmasss}}%   m:291 n:291 o:  p:  q:
        \divide \value{papermas@tmpn} by 100%                 m:297 n:2   o:  p:  q:
```

```
\newcounter{papermas@tmpo}
\setcounter{papermas@tmpo}{\arabic{papermas@tmpn}}%   m:291 n:2   o:2 p:  q:
\multiply \value{papermas@tmpn} 10%                   m:297 n:20  o:2 p:  q:
\divide \value{papermas@tmpm} by 10%                  m:29  n:20  o:2 p:  q:
\newcounter{papermas@tmpp}
\setcounter{papermas@tmpp}{\arabic{papermas@tmpm}}
\addtocounter{papermas@tmpp}{-\arabic{papermas@tmpn}}%m:29  n:20  o:2 p:9 q:
%        29              - 20 = 9
\multiply \value{papermas@tmpm} 10%                   m:290 n:20  o:2 p:9 q:
\newcounter{papermas@tmpq}
\setcounter{papermas@tmpq}{\arabic{papermasmasss}}
\addtocounter{papermas@tmpq}{-\arabic{papermas@tmpm}}%m:290 n:20  o:2 p:9 q:7
%       297             - 290 = 7
```

Now rounding mathematically correct, i. e. ≥ 0.5 becomes 1 (and remember a possible amount carried forward!) and < 0.5 becomes 0.

```
\ifnum\value{papermas@tmpq}>4
  \addtocounter{papermas@tmpp}{1}%                    m:290 n:20 o:2 p:10 q:7
  \ifnum\value{papermas@tmpp}>9%                      m:290 n:20 o:2 p:10 q:7
    \addtocounter{papermas@tmpo}{1}%                  m:290 n:20 o:3 p:10 q:7
    \setcounter{papermas@tmpp}{0}%                    m:290 n:20 o:3 p:0  q:7
  \fi
\fi
```

The result in the example above is $297/100 = 2.97 \approx 3.0$. We write this into `\papermastmpr` (where `\papermas@decimalsep`) is the decimal separator) and the (other) options' values into temporary definitions, as well as the number of sheets:

```
\edef\papermastmpr{\arabic{papermas@tmpo}\papermas@decimalsep\arabic{papermas@tmpp}}%
\xdef\papermas@mbs{\arabic{papermas@tmpo}}%
\edef\papermastmpformat{\papermas@format}%
\edef\papermastmpmasss{\papermas@masss}%
\edef\papermastmppagespersheet{\papermas@pagespersheet}%
\edef\papermastmpt{\arabic{papermas@sheets}}%
```

We use the pageslts package, which already was used to determine the total number of pages, to check for the counter `papermassttl`. If it exists, nothing is done, if it does not exist, it is declared as `\newcounter` (and by default set to zero).

```
\pagesLTS@ifcounter{papermassttl}
```

If the `papermassttl` counter value already has the value of `papermasmasss`, everything is fine.

```
\ifnum\value{papermassttl}=\value{papermasmasss}
  \relax
```

Otherwise we need another run of LaTeX.

```
\else
  \AtEndAfterFileList{%
    \PackageWarningNoLine{papermas}{%
      Number of pages may have changed.\MessageBreak%
      Rerun to get it right%
     }%
    }%
\fi
```

In any case, we set the counter `papermassttl` to the current value of `papermasmasss`.

```
\setcounter{papermassttl}{\arabic{papermasmasss}}
```

Because we want to write out into the aux-file, we need the expanded value (as string) of papermasmasss:

```
\edef\papermastmps{\arabic{papermasmasss}}%
```

If we are allowed to write into the aux-file, we do it here. If we are not allowed to do it, the pageslts package already gave an according error message.

```
\if@filesw%
```

When it is read from the aux-file and when its content is processed, the counter papermassttl might not have been defined yet. Therefore we again use the \pagesLTS@ifcounter command of the pageslts package.

```
  \immediate\write\@auxout{\string
    \pagesLTS@ifcounter{papermassttl}}%
```

We set the counter papermassttl to the value \papermastmps,
i. e. \arabic{papermasmasss}. In the next compilation run, it will be checked, whether \value{papermassttl}=\value{papermasmasss} (see above).
If this is the case, everything is OK, no changes happened, and no rerun is necessary (at least not for papermas).

```
  \immediate\write\@auxout{\string
    \setcounter{papermassttl}{\papermastmps}}%
```

What we do need, is to get the determined \papermastmpr to the user. Therefore

1. we define \papermasstotal in the aux-file, where the user can look it up
2. we define \papermasstotal, so the user can e. g. write

```
This document consists of $\arabic{pagesLTS.pagenr}$~pages.
When printing $\papermaspagespersheet$~pages on one sheet of
paper, $\papermassheets$~sheets will be needed. For
ISO~A~\papermasformat\ paper of $\papermasmasss\unit{g}\unit{m}^{-2}$
specific mass, the printout will have a mass of about
$\papermasstotal\unit{g}$.
```

```
      \immediate\write\@auxout{\string
        \gdef\string\papermasstotal{\papermastmpr}}%
      \immediate\write\@auxout{\string
        \gdef\string\papermasformat{\papermastmpformat}}%
      \immediate\write\@auxout{\string
        \gdef\string\papermasmasss{\papermastmpmasss}}%
      \immediate\write\@auxout{\string
        \gdef\string\papermaspagespersheet{\papermastmppagespersheet}}%
```

3. we give at the screen the information about the \papermasstotal (see \AtEndAfterFileList below)
4. which will also appear in the log-file.

We want to give also `\papermastmpt = \arabic{papermas@sheets}` to the user, i. e. the number of sheets needed to print the document. Therefore we follow the same procedure:

```
          \immediate\write\@auxout{\string
            \gdef\string\papermassheets{\papermastmpt}}%
        \fi%
      \fi%
    \fi%
  \fi%
  }
```

`\AtBeginDocument` `\AtBeginDocument` it is checked whether some commands, which are/will be defined via the aux-file, are undefined yet. If this is the case, `\AtEndAfterFileList` a rerun warning is given.

```
\AtBeginDocument{%
  \gdef\papermas@undefined{\textbf{??}}
  \gdef\papermas@rerun{0}
  \ifx\papermasstotal\papermas@undefined        \gdef\papermas@rerun{1} \fi
  \ifx\papermasformat\papermas@undefined        \gdef\papermas@rerun{1} \fi
  \ifx\papermasmasss\papermas@undefined         \gdef\papermas@rerun{1} \fi
  \ifx\papermaspagespersheet\papermas@undefined \gdef\papermas@rerun{1} \fi
  \ifx\papermassheets\papermas@undefined        \gdef\papermas@rerun{1} \fi
  \ifx\papermas@rerun\pagesLTS@one
    \AtEndAfterFileList{
      \PackageWarningNoLine{papermas}{%
        Variable(s) still undefined!\MessageBreak%
        Rerun to get the variable(s) right%
       }
     }
  \fi
  }
```

`\AfterLastShipout` What we did not do yet, is to really *call* the command `\papermas@totmass`. We do this `\AfterLastShipout`, because we need the total number of pages, and asking for them at the end of the document might save another compilation run.

```
\AfterLastShipout{%
  \papermas@totmass%
  }%
```

`\AfterLastShipout` is a command from the atveryend package of Heiko Oberdiek, which is already loaded by the pageslts package (about how to get the atveryend package, please see the documentation of the pageslts package – you may need to get further packages for pageslts anyway, if they have not been installed within your LaTeX system).

For pretty printing the message of papermas three internal commands are needed. We borrow the pagesLTS.pnc.0 counter from the pageslts package instead of defining another new one.

```
\newcommand{\papermas@log}[1]{%
  \ifnum#1>9%
    \addtocounter{pagesLTS.pnc.0}{1}%
    \papermas@log{\intcalcDiv{#1}{10}}%
  \fi%
  }

\newcommand{\papermas@spaces}[2]{%
  \edef\papermas@remember{\arabic{pagesLTS.pnc.0}}%
  \setcounter{pagesLTS.pnc.0}{1}%
  \papermas@log{#1}%
  \addtocounter{pagesLTS.pnc.0}{-#2}%
  \multiply \value{pagesLTS.pnc.0} -1%
  \papermas@space{\arabic{pagesLTS.pnc.0}}%
  \message{*^^J}%
  \setcounter{pagesLTS.pnc.0}{\papermas@remember}%
  }

\newcommand{\papermas@space}[1]{%
  \ifnum \value{pagesLTS.pnc.0}>0%
    \message{}%
  \fi%
  \setcounter{pagesLTS.pnc.0}{#1}%
  \addtocounter{pagesLTS.pnc.0}{-1}%
  \ifnum \value{pagesLTS.pnc.0}>0%
    \papermas@space{\arabic{pagesLTS.pnc.0}}%
  \fi%
  }

```

\AtEndAfterFileList

```
\AtEndAfterFileList{%
```

\AtEndAfterFileList{...} is even later than \AfterLastShipout:

> "This code is called right before the final \cs{@@end}."

(atveryend package of Heiko Oberdiek, v1.6 as of 2011/04/15).

If no necessarity for a rerun was *detected* (Check for other rerun warnings!), the final \PackageInfo is given.

```
  \ifx\papermas@rerun\pagesLTS@zero%
    \message{^^J}%
    \message{papermas: ******************** Paper mass ********************^^J}%
    \message{papermas: * This document consists of \arabic{pagesLTS.pagenr} pages.}
    \papermas@spaces{\arabic{pagesLTS.pagenr}}{16}%
    \message{papermas: * When printing \papermaspagespersheet\space pages on one sheet of pape
    \papermas@spaces{\papermaspagespersheet}{6}%
    \message{papermas: * \papermassheets\space sheets will be needed.}
    \papermas@spaces{\papermassheets}{26}%
    \message{papermas: * For ISO A \papermasformat\space paper of \papermasmasss\space g/m^2 sp
    \papermas@spaces{\papermasmasss}{7}%
    \message{papermas: * the printout will have a mass of about \papermasstotal\space g.}
    \papermas@spaces{\papermas@mbs}{5}%
    \message{papermas: **************************************************^^J}
```

```
    \message{^^J}
  \fi%
  }
```

`\papermas@powerof` The command `\papermas@powerof` is **obsolete**. `\intcalcPow` is used instead. For compatibility reasons we still provide the command (but with other code), and issue an error message.

```
\newcommand\papermas@powerof[2]{%
  \PackageError{papermas}{Obsolete command \string\papermas@powerof\space used}{%
    The command \string\papermas@powerof\space has been removed from the papermas package.\Mess
    Please use e.g. \string\intcalcPow\space from the intcalc package instead.\MessageBreak%
    You can now just type Return to continue, but this error message will be\MessageBreak%
    issued again when using \string\papermas@powerof,\space and the command might be\MessageBr
    removed completely from future versions of the papermas package.\MessageBreak%
   }%
  \AtEndAfterFileList{%
    \message{^^J%
      papermas: Please remember to replace the \string\papermas@powerof\space command!^^J^^J%
     }%
   }%
  \pagesLTS@ifcounter{papermas@result}%
  \setcounter{papermas@result}{\intcalcPow{#1}{#2}}%
  }
```

⟨/package⟩

6 Installation

6.1 Downloads

Everything is available at CTAN:, http://www.ctan.org/tex-archive/, but may need additional packages themselves.

papermas.dtx

For unpacking the papermas.dtx file and constructing the documentation it is required:

- TEXFormat LATEX 2ε: http://www.CTAN.org/
- document class ltxdoc, 2007/11/11, v2.0u,
  CTAN:macros/latex/base/ltxdoc.dtx
- package holtxdoc, 2011/02/04, v0.21,
  CTAN:macros/latex/contrib/oberdiek/holtxdoc.dtx
- package hypdoc, 2010/03/26, v1.9,
  CTAN:macros/latex/contrib/oberdiek/hypdoc.dtx

papermas.sty

The papermas.sty for LATEX 2ε (i. e. all documents using the papermas package) requires:

- TEXFormat LATEX 2ε, http://www.CTAN.org/
- package intcalc, 2007/09/27, v1.1,
  CTAN:macros/latex/contrib/oberdiek/intcalc.dtx
- package kvoptions, 2010/12/23, v3.10,
  CTAN:macros/latex/contrib/oberdiek/kvoptions.dtx
- package pageslts, 2011/08/08, v1.2a,
  CTAN:macros/latex/contrib/pageslts/pageslts.dtx

papermas-example.tex

The papermas-example.tex requires the same files as all documents using the papermas package, and additionally:

- class article, 2007/10/19, v1.4h, from classes.dtx:
  CTAN:macros/latex/base/classes.dtx
- package papermas, 2011/08/22, v1.0h,
  CTAN:macros/latex/contrib/papermas/papermas.dtx
  (Well, it is the example file for this package, and because you are reading the documentation for the papermas package, it can be assumed that you already have some version of it – is it the current one?)

totpages

As possible alternative in section 3 there is listed

- package totpages, 2005/09/19, v2.00,
  CTAN:macros/latex/contrib/totpages/totpages.dtx

Oberdiek
holtxdoc
atveryend
intcalc
kvoptions

All packages of Heiko Oberdiek's bundle 'oberdiek' (especially holtxdoc, atveryend, intcalc, and kvoptions) are also available in a TDS compliant ZIP archive: CTAN:install/macros/latex/contrib/oberdiek.tds.zip.
It is probably best to download and use this, because the packages in there are quite probably both recent and compatible among themselves.

hyperref

hyperref is not included in that bundle and needs to be downloaded separately, http://mirror.ctan.org/install/macros/latex/contrib/hyperref.tds.zip.

Münch

A hyperlinked list of my (other) packages can be found at http://www.Uni-Bonn.de/~uzs5pv/LaTeX.html.

6.2 Package, unpacking TDS

Package. This package is available on CTAN:.

CTAN:macros/latex/contrib/papermas/papermas.dtx
: The source file.

CTAN:macros/latex/contrib/papermas/papermas.pdf
: The documentation.

CTAN:macros/latex/contrib/papermas/papermas-example.pdf
: The compiled example file, as it should look like.

CTAN:macros/latex/contrib/papermas/README
: The README file.

CTAN:install/macros/latex/contrib/papermas.tds.zip
: Everything in TDS compliant, compiled format.

which additionally contains

papermas.ins	The installation file.
papermas.drv	The driver to generate the documentation.
papermas.sty	The `.sty`le file.
papermas-example.tex	The example file.

For required other packages, see the preceding subsection.

Unpacking. The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain TEX:

```
tex papermas.dtx
```

About generating the documentation see paragraph 6.4 below.

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
papermas.sty          → tex/latex/papermas.sty
papermas.pdf          → doc/latex/papermas.pdf
papermas-example.tex  → doc/latex/papermas-example.tex
papermas-example.pdf  → doc/latex/papermas-example.pdf
papermas.dtx          → source/latex/papermas.dtx
```

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

6.3 Refresh file name databases

If your TEX distribution (teTEX, mikTEX,...) relies on file name databases, you must refresh these. For example, teTEX users run `texhash` or `mktexlsr`.

6.4 Some details for the interested

Unpacking with LaTeX. The `.dtx` chooses its action depending on the format:

plain TeX: Run docstrip and extract the files.

LaTeX: Generate the documentation.

If you insist on using LaTeX for docstrip (really, docstrip does not need LaTeX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{papermas.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by a configuration file `ltxdoc.cfg`. For instance, put this line into that file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLaTeX:

```
pdflatex papermas.drv
makeindex -s gind.ist papermas.idx
pdflatex papermas.drv
makeindex -s gind.ist papermas.idx
pdflatex papermas.drv
```

6.5 Compiling the example

The example file, papermas-example.tex, can be compiled via
`latex papermas-example.tex`
or (recommended)
`pdflatex papermas-example.tex`
but will need probably three compiler runs to get everything right.

7 Acknowledgements

I would like to thank Heiko Oberdiek (heiko dot oberdiek at googlemail dot com) for providing a lot (!) of useful packages (from which I also got everything I know about creating a file in `.dtx` format, ok, say it: copying), and the news:comp.text.tex and news:de.comp.text.tex newsgroups for their help in all things TeX.

8 History

[2010/06/01 v1.0(a)]

- First version of this papermas package.

[2010/06/03 v1.0b]

- New `\papermassheets` and reruncheck introduced; several small changes.
- Example adapted to other examples of mine.
- Updated references to other packages.
- TDS locations updated.
- Several changes in the documentation and the Readme file.

[2010/06/24 v1.0c]

- holtxdoc warning in `drv` updated.
- Corrected the location of the package at CTAN.
  (TDS was still missing due to packaging error.)
- Updated references to other packages: hyperref and pagesLTS.
- Added a list of my other packages.
- Several changes to the documentation.
- Introduced new **option**: `decimalsep`.

[2010/07/29 v1.0d]

- Corrected given url of `papermas.tds.zip` and other urls.
- There is a new version of the used hyperref package: 2010/06/18, v6.81g.
- There is a new version of the used pagesLTS package: 2010/07/29, v1.1e.
- Included a `\CheckSum`.

[2011/02/01 v1.0e]

- Updated to version 2010/12/16 v6.81z of the hyperref package.
- Removed wrong % from the driver file.
- Changed the `\unit` definition (got rid of an old `\rm`).
- Replaced the list of my packages with a link to a web page list of those, which has the advantage of showing the recent versions of all those packages.
- Now using `\@ifundefined`.
- Removed `/muench/` from the path at diverse locations.
- There is a new version of the used pagesLTS package: 2011/02/01, v1.1m.
- Some small changes.

[2011/06/02 v1.0f]

- There is a new version of the used kvoptions package: 2010/12/23, v3.10.
- There is a new version of the used pagesLTS package: 2011/03/17, v1.1o.
- The holtxdoc package was fixed (recent version: 2011/02/04, v0.21), therefore the warning in `drv` could be removed. – Adapted the style of this documentation to new Oberdiek `dtx` style.
- There is a new version of the used hyperref package: 2011/04/17, v6.82g.
- The rerun warnings are given after the `filelist` (if that is called with `\listfiles`) and the final papermas information is presented `\AtVeryVeryEnd` (now only ones instead of twice).
- Replaced `\text` by `\textrm`.
- Instead of compiling "a to the power of b" itself, papermas now uses the intcalc package of Heiko Oberdiek.
- Removed five counters.
- A lot of small changes (also in the README).

[2011/08/08 v1.0g]

- The pagesLTS package has been renamed to pageslts: 2011/08/08, v1.2a.
- Replaced `\global\edef` by `\xdef`.
- Minor details.

[2011/08/22 v1.0h]

- Hot fix: TeX 2011/06/27 has changed `\enddocument` and thus broken the `\AtVeryVeryEnd` command/hooking of atveryend package as of 2011/04/23, v1.7. Until it is fixed, `\AtEndAfterFileList` is used.

When you find a mistake or have a suggestion for an improvement of this package, please send an e-mail to the maintainer, thanks! (Please see BUG REPORTS in the README.)

9 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.